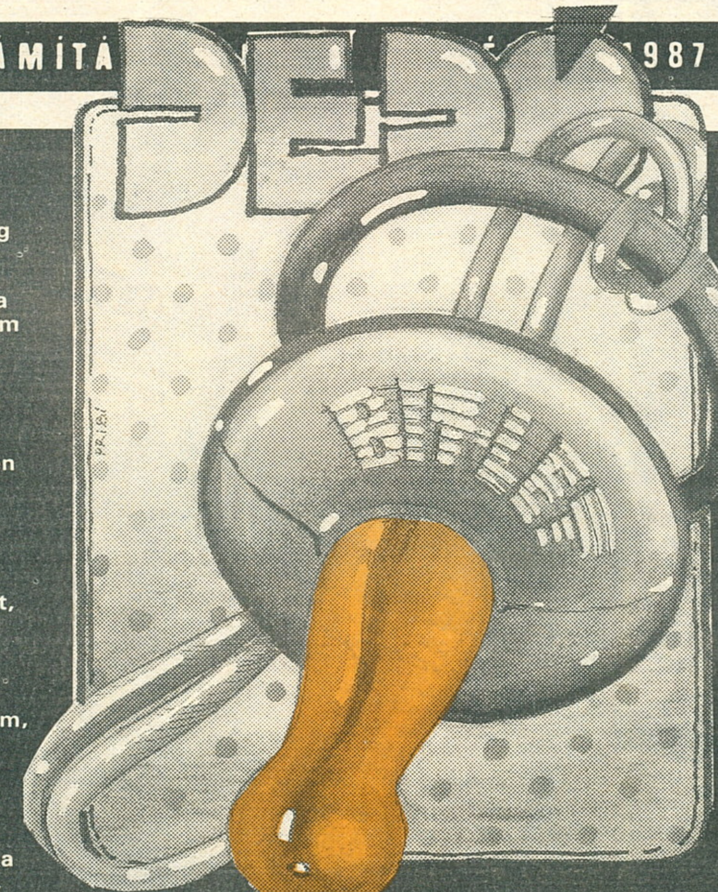
BIT-LET



Néhány héttel ezelőtt egyik barátom jóindulatúlag figyelmeztetett: ifjú ismerősei kérték, mondaná meg nekem, mint a BIT-LET szerkesztőjének, hogy szerintük mostanában lapocskánk mintha átment volna dedóba. Tartottam hát rögvest egy kis ön- vagy inkább lapvizsgálatot, valóban így van-e? Mielőtt bármit leírnék az önvizsgálat eredményeiről, hadd jelezzem az olvasóknak, hogy természetesen mindig, minden kritikát szívesen fogadunk, s persze ebben a konkrét kérdésben is örülnénk, ha minél több vélemény birtokában hozhatnánk meg bizonyos döntéseket. Mert, döntések is lesznek, hogy milyenek, arról később. Nos, az önvizsgálat váratlan eredménnyel zárult. Átnézve az elmúlt hónapok lapszámait, nekem, a lap szerkesztőjének úgy tűnt, mintha túlságosan is magasan lenne a mérce. Régóta vágyam már, hogy a lapban több helyet kellene szorítani a kezdőknek, azoknak, akik nem a matematika felől érkeztek a számítógéphez, meg azoknak, akik épp csak játszogatnak vele, meg nagynéha bepötyögnek egy-egy programot. Mintha mostanában a mi lapunk is átment volna szakmaiba. A cikkek jelentős része azoknak szól, akik egyik vagy másik géppel már régebben eljegyezték magukat, s minél mélyebb ismeretekre vágynak. Való igaz, hogy nekik dedós színvonalú egyik-másik levelezőnk kérdése, meg az arra adott válasz, sőt dedós lehet sok vájtfülű olvasónk számára például a Sorvezető. Nos, ez az a rovat, amellyel kapcsolatban döntés előtt állunk, s ebben a döntésben szívesen hagyatkoznánk olvasóink véleményére is. Nincs ugyanis semmiféle visszajelzésünk arról, hogy a lapnak ezt a rovatát akár egyetlen pedagógus is fölhasználja-e? Ez a rovat ugyanis tipikusan azok közé tartozik, amelyet nagyon sok olvasónk átlapoz. Ez különösebben nem zavar bennünket, föltéve, hogy valóban eléri a rovat a célját és segíti a szakkörök munkáját. Ezúton kérjük tehát a nagyrabecsült oktatással foglalkozó olvasóinkat, hogy mondjuk egy-egy szavas levélkében, táviratban stb. jelezzék, hogy kell-e nekik ez a rovat vagy sem. (Ha semmilyen levélkét nem kapunk, ígérjük, hogy nem foglalja a helyet tovább ez a rovat.) Másfelől azonban szeretnénk a „dedós" vádakkal kapcsolatban már jóelőre közölni, hogy szeptembertől kezdve leszünk azután igazán dedósak, szeretnénk ugyanis beindítani a „beszállókártya" című rovatunkat, amelyben, ha nem is programozói tanfolyamot, de kezdőknek szóló tanácsokat közölnénk. A dedó ellen tiltakozó olvasók persze mindig a jelmondatunkra hivatkoznak, csak azt felejtik el, hogy azok is olvasóink, s azokért is van a szerkesztő, akik épp azért nem veszik kezükbe a BIT-LET-et, mert nekik kínaiul van. Apropos jelmondat. Hogy mennyire komolyan vesszük, azt láthatják a 31. oldalon. Több pályázónk szóvá tette ugyanis, hogy ha kivágja a pályázati szelvényt, akkor gyakran valami fontosba vág bele. Kétségtelen, hogy az utóbbi időben egyre kevésbé figyeltünk erre, minthogy kevés pályázónk volt, s így keveseket érintett a dolog. Most az Atari nyerőre azonban soha nem látott menynyiségben érkeztek a megoldások, s így megérkeztek a panaszos levelek is. Nos, mostantól kezdve ügyelünk a dologra. A dedóról – mert hiszen ezzel kezdtük – befejezésül pedig annyit: lehet, hogy van, aki úgy véli: lementünk dedóba, mi viszont úgy véljük, hogy az lenne az igazi számítástechnikai lap, amely cikkeinek nagyobbik részét e kezdők is értenék, de a vájtfülűek is találnának benne új információkat. Föltéve persze ha elolvasnák. Nem állítjuk, hogy nekünk sikerült ilyen lapot csinálni, de abban biztosak vagyunk, hogy a dedós cikkeket is érdemes átfutni. Hátha kiderül, hogy nem is olyan dedósok.

Angyalosi László

BELÜLRŐL

ötlet

HÍROLDAL



PROMETHEUS

A címbeli néven hozott létre új kutatási programot a nyugat-európai országok Euréka elnevezésű fejlesztési programjában tizenhárom autógyár és negyven kutatóintézet. A következő években több mint egymilliárd nyugatnémet márkát költenek egy autó fedélzeti computer kifejlesztésére. A személygépkocsikba szerelendő, elképzelt számítógép ellenőrizné a gépjármű működését, szabályozná az üzemanyag-felhasználást, a káros szennyezőanyagok kibocsátását, védene a külső veszélyforrásoktól, és folyamatosan ellátná a vezetőt a szükséges közlekedési információkkal.

NÉPZENE

Népzenét elemző kottás adattárat kívánnak létrehozni – számítógép segítségével – az Állami Gorkij Könyvtárban. A program kidolgozását négy évvel ezelőtt kezdték el azzal a céllal, hogy tízezer magyar népdalt számítógépre vigyenek. A programot könyvtárak, valamint ének-zenét oktató intézmények használhatják majd, eredményesebbé tehetik vele a fiatalok zenei nevelését. Az úgynevezett NEKTÁR program lehetőséget ad arra, hogy a népdalokat mágneslemezen tárolják, majd a kottát képernyőre kiírva elemezzék, és a dallamot is ismételhetően lejátsszák. A dalokat különböző jellemzők szerint lehet megkeresni. Így csoportosíthatók – többek között – a forrás, a kezdő sor, a gyűjtő neve, a gyűjtés helye és ideje, a stílus, az előadásmód, a szótagszám, a sorszerkezet és a hangterjedelem szerint. Igaz, a Commodore–64 „hangja" nem olyan, mintha hangszerek szólaltatnák meg a népdalokat, de a hangmagasság és a ritmus minden esetben pontos.

ASCA

A mesterséges intelligencia kutatás, az ötödik generációs számítógép-fejlesztési program legújabb állomása Japánban az ASCA elnevezésű számítógép megalkotása. A gyorsaságban minden eddigieket mintegy százszorosan felülmúló gép asszociatív úton alkalmas a problémák elemzésére, feltételezések kimunkálására és a szükséges következtetések levonására. Az új szupergépet a japán távíró- és telefontársaság kutatói fejlesztették ki.

FÖLDRENGÉS

Régi vágya az embernek, hogy valamilyen módon előre jelezhesse a földrengések bekövetkezését. Különösen így van ez Japánban, ahol évente mintegy kétezer kisebbnagyobb földrengés következik be. Ezért kezdett közösen számítógépes földrengés-előrejelző rendszert kifejleszteni a tokiói egyetem földrengéskutató intézete, a meteorológiai intézet és a Mitsubishi kutatóintézet. A számítógépes rendszer három gépből áll: a földrengésekkel kapcsolatos adatokat tárolóból és elemzőből, az elméleti földrengés-kutatásokat feldolgozóból és a rendvetkeztetések levonását végzőből. A kéregmozgások szer tartalmazza, például a kéregmozgások adatait és a talajvíz, illetve a föld alatti karsztvizek szintjének változásait, évekre visszamenően. A kutatók azt remélik, hogy a földrengések „törvényszerűségeiből" kiindulva előre tudják jelezni a következő földrengések epicentrumát, erősségét és bekövetkezésének időpontját.

-CLN

COMMODORE-MEZBEN

Európa egyik legjobb klubcsapata, a Dinamó Kijev a közelmúltban Budapesten vendégszerepelt. Ekkor tudtuk meg, hogy a megelőző kupamérkőzésén a Dinamó játékosai Commodore feliratú mezben futballoztak. Kérdésekre válaszolva Lobanovszkij edző elmondta, hogy az angol céggel a reklámüzletet a szovjet sporthivatal kötötte, és a jogdíjak is nekik folynak be.

CLEAN ROOM

Azok a terhes nők, akik a precízen szabályozott hőmérsékletű és páratartalmú rendkívül pormentes, chipgyártó helyiségekben (cleam room) dolgoznak, azt kockáztatják, hogy könnyen elvetélhetnek – állapították meg amerikai kutatók. A University of Massachusetts tudósai megállapították, hogy az ilyen extrasteril körülmények között dolgozó terhes nők 39 százaléka vetél el (az átlagos szám 20 százalék). A kutatók most azt kezdték el vizsgálni, hogy a „tiszta szoba" milyen hatással van az egészséges, nemzőképes férfiakra és nőkre.

IBM SZUPRAVEZETŐ

Újabb szupravezetési rekordról érkezett hír: ezúttal az IBM amerikai számítógépkonszern jelentette be, hogy „áttörésnek tekinthető sikert" ért el a szupravezetésért – azaz az elektromos ellenállás nélküli vezetőkért – vívott versenyben. Állítólag a korábbinál sokkal jelentősebb áramerősséget képes az általa előállított különleges anyag ellenállás nélkül továbbítani. Az IBM közlése szerint laboratóriumában olyan anyagot állított elő, amely egy négyzetcentiméter keresztmetszeten 100 ezer amper erősségű áramot továbbít – veszteség nélkül. Igaz, a hőmérséklet, amelyen az IBM eredményét elérte, még mindig meglehetősen alacsony: minusz 195 fok. De ez mégis viszonylag kedvező, mert hélium helyett folyékony nitrogén lehet a hűtőközeg. Ez utóbbi sokkal olcsóbban előállítható, mint a hélium.

FANTOMOK

Egy NSZK-beli összeszerelő üzemben induktív vezérlésű, vezető nélküli „fantomjárműveket" használnak. A központi számítógép által vezérelt járművek mozgó szerelőállomásokként működnek és helyettesítik a szállítószalagot. A járműveket megrakják az adott részegységekkel és elküldik a kijelölt munkahelyre. Ezt a szerelés és az elkészült egység visszaszállítása követi. Az induktív járművek alkalmazása segíti a vállalatot a rugalmas, gyors termékváltásban és az ergonómiai szempontok tökéletesebb érvényesítésében.

HÁROMDIMENZIÓS

Térhatású mozi, térhatású televízió kifejlesztésén régen dolgoznak a kutatók az egész világon. Mint tudjuk, átütő sikerű, a gyakorlatban is használható, olcsó megoldást még nem sikerült találni. Talán most sikerülni fog. Három japán cég, a Sharp, a Matsushita és a JVC kutatói olyan különleges szemüveg kidolgozásán munkálkodnak, amelyen át egy színes televíziós adás térhatásúként érzékelhető. A különleges szemüveg lencséi helyén folyadékkristályoslapokat helyeztek el. A lapokat a televízió képjelimpulzusai úgy szinkronizálják, hogy amikor a „bal oldali" kép jelenik meg a képernyőn, akkor a szemüveg bal oldala átlátszó, és a „jobb oldali" kép esetén a jobb oldala. A szem tehetetlensége miatt a két képet az ember térhatású képként érzékeli.

SAKKBANK

Érdekes bemutatóra került sor a közelmúltban a Novotrade RT Játékstúdiójában. Az NSZK-beli Frederic Friedel sakkszakíró és munkatársa, Mathias Wüllenweber számítógép-programozó számítógépes sakkadatbankjukat ismertették meg a szakmai közönséggel. Az Atari számítógépprogram segítségével több ezer sakkjátszma közül pillanatok alatt visszakereshető bármelyik mérkőzés; lépésről lépésre le is játszható. A szakemberek véleménye szerint ez az adatbank forradalmasíthatja az élversenyzők felkészülését, hozzájárulhat a sakkelmélet rohamos fejlődéséhez.

NDK CSÚCS-TECHNOLÓGIA

A Német Demokratikus Köztársaság gazdasága ma elsősorban a csúcstechnológiák fejlesztésére összpontosítja erőforrását. A mikroelektronika, a számítástechnika és a robottechnika területén saját, ütőképes termelőbázist hoztak létre. Ezek az iparágak a jelenlegi ötéves tervben nemcsak viszik a pénzt, hanem egyre inkább elősegítik a termelés korszerűsítését, és hatásuk érezhetővé válik az egész gazdaságban. Az idén több mint tizennégyezer ipari robotot állítanak munkába, míg terveik szerint a tervidőszak végére nyolcvanezer robotot állítanak elő.

ZSEBMÁSOLÓ

Panacopy néven kisméretű, elemes zsebmásológépet hozott forgalomba a japán Matsushita cég. A kis készülékbe infravörös sugarakat emittáló diódákat, fényérzékelőket, és hőnyomtatókat építettek be. A zsebmásoló hőérzékeny papírtekercsre írja, illetve rajzolja a másolandó szöveget és grafikonokat. A másológépet bekapcsolása után a másolandó dokumentumra kell helyezni és azon lassan végighúzni. Eközben az érintkezési felületeken a fényt emittáló diódák megvilágítják a dokumentumot, a szenzorok pedig érzékelik a szavak és ábrák világosabb és sötétebb árnyalatait, amit elektromos jelek alakjában továbbítanak a nyomtatóhoz. Ez utóbbi az eredeti fekete-fehér árnyalatokkal égeti be a hőérzékelő papírba a szavakat és ábrákat. Várható, hogy a zsebmásológépek a közeljövőben mint távmásolók is használhatók lesznek. A Matsushita mérnökei ugyanis egy interface megvalósításán dolgoznak, amely lehetővé teszi a Panacopy csatlakozását telefonvonalhoz vagy számítógéphez, hogy a digitális jelek továbbíthatók legyenek.



ÚJ!

PowerMate 1

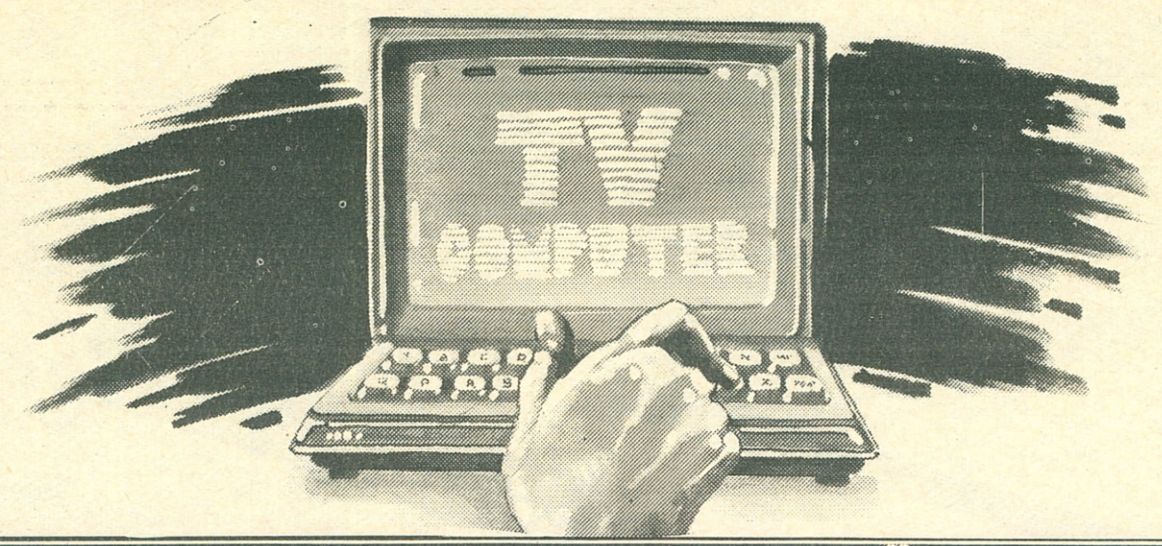
A japán NEC cég bemutatta három új, 80286 processzorra épült, teljes egészében IBM PC AT kompatibilis személyi számítógép típusát. A PowerMate 1 a legolcsóbb változat, 8 MHz-es processzor sebességgel, 640 Kbyte-os RAM-mal, ami 8,6 Mbyte-ig bővíthető. A rendszer magába foglal két 5$^1/_4$-es floppyt és egy 3$^1/_2$-es lemezmeghajtót. A PowerMate 2 már átkapcsolhatóan 8 vagy 10 MHz sebességű. RAM-ja 10,5 Mbyte-ig bővíthető. A harmadik változat a BusinessMate, egy nagyteljesítményű, többfelhasználós rendszer, melyhez nyolc terminál csatlakoztatható. RAM-ja 10,6 Mbyte-ig bővíthető, egy 1,2 Mbyte-os floppymeghajtót és egy 40 Mbyte kapacitású merevlemezes meghajtót tartalmaz.

KÉPTELEFON

A képtelefon nem a sci-fik látványossága többé: már rendelkezésre állnak a megfelelő mikroprocesszorok, a gyors jelprocesszorok, az adóban és a vevőben pedig 3–4 képtároló. Így már csak egy megfelelően nagy integráltságú chip kidolgozása szükséges a minél kedvezőbb ár eléréséhez. Ez az NSZK-beli AEG jelenlegi programja. 1988-tól részt vesznek a szövetségi posta ISDN próbaüzemi kísérleteiben, 1990-ig elvégzik a szükséges szabványosítást, ezután pedig már semmi sem állhat a képtelefon bevezetésének útjába.

NAPERŐMŰ

Számítógép vezérli a Krím-félszigeten most felépült szovjet naperőművet. A számítógépes vezérlőrendszer irányítja a negyven négyzetméter összfelületű, húsz gyűrűben elhelyezkedő hatszáz tükröt, amelyekről egy 89 méter magas energiatermelő torony kazánjába verődnek a napsugarak. A kazánban gőzt állítanak elő, ami az erőmű turbináit hajtja. A Szovjetuniónak ezen a vidékén a napsütéses órák száma eléri az évi 2320-at. A termelt energia egy részét akkumulátorok tárolják. A kísérleti jellegű állomás valamennyi berendezése a Szovjetunióban készült.



ELSŐ KÉZBŐL

A TV COMPUTERRŐL

AZ EDITORRÓL

A TV-Computer képernyős szerkesztővel rendelkezik, ennek szokásos angol megnevezése „full screen editor", vagy röviden csak editor. A BASIC parancsállapotában az editor az elsődleges beviteli eszköz, bekapcsolás után először ezzel találkozunk. A TVC Kezelési Útmutató 2. fejezete lényegében ennek ismertetésével foglalkozik. Ezek a rutinok azonban bármely felhasználói program számára is elérhetők a szokásos RST 30h hívással.
Az editor szoros kapcsolatban áll a video és a billentyűzet rutinokkal, melyeket már korábban ismertettünk. Például az input rutin szabványos billentyűzet karakter input hívást tartalmaz (RST 30h, KBD–CHIN), az output pedig a video rutinokkal közösen használt képernyőre ír. Ez utóbbi tulajdonság azt is megerősíti, hogy csak egyféle képmegjelenítés van, tehát nincs külön rajzoló és szöveges mód.

AZ EDITOR ÁLTALÁNOS ISMERTETÉSE

Az editor a képernyőn 24 karaktersort kezel. Az egy sorba írható karakterek száma a grafikus felbontástól függ: kétszínű módban 64, négyszínű módban 32, tizenhat színű módban pedig 16. A grafikus mód váltása (VMODE video rutin hívással vagy a GRAPHICS paranccsal) az editort is alaphelyzetbe állítja. Az aktuális tintaszín és háttérszín azonos azzal, amit a video rutinok használnak (sINK és sPAPER változók), a karakterek kiírása azonban nem a video rutinnal, hanem közvetlenül történik. A 8×10 pontból álló karakter minden pontja kiírásra kerül, tehát felülírja az eredeti képernyőtartalmat.
A szerkesztési műveletekhez szükség van a képernyőre írt karakterek kódjának, az ASCII kódnak az eltárolására is. Ez a karakteres memória (ASCII–MAP) a 256=100h címen kezdődik, és 25 sornak egyenként 64 byte helyet, összesen 1600 byte-ot foglal le. A 25. sort az editor nem használja, az ehhez tartozó 64 byte szabadon felhasználható. A soroknak mindig az eleje az aktív, a felbontástól függően az első 16, 32 vagy 64 byte. Csak a megjeleníthető karakterek kódjait (32–223) írja ide az editor, a vezérlőkarakterek esetén a megfelelő funkciót hajtja végre.
Megjegyezzük, hogy az ASCII–MAP területére beírt karakterkód hatására még nem jelenik meg a karakter a képernyőn, sőt a beírt kód csak akkor lesz befolyással az editorra, ha olyan helyre írtuk, ahol eddig is volt karakter.
A kurzor csak a karakter input idején látszik, de akkor mindig, tehát nem tiltható le. A villogás üteme állandó, nem módosítható. A villogtatás során a kurzor képe és az eredeti képernyőtartalom felváltva látszik. Ha nem teszünk mást, csupán a kurzort mozgatjuk, akkor a képernyőn látható rajz vagy szöveg nem fog megváltozni.
Négyféle kurzor karakter fordulhat elő, az aktuális LOCK módnak, illetve a billentyűzet rutinhoz tartozó LOCK–KEY változónak megfelelően:

téglalap (7Fh) normál mód (LOCK)
inverz C (9Eh) nagybetűk (LOCK+CTRL)
inverz S (9Fh) folyamatos shift (LOCK+SHIFT)
inverz A (8Fh) alternatív karakterek (LOCK+ALT)

Az editor jellemzője, hogy képes több összetartozó képernyősort egyetlen logikai sornak, „bekezdés"-nek tekinteni. A szerkesztési műveletek helyét a kurzor pozíciója jelöli ki, a műveletek azonban mindig a bekezdésre vonatkoznak. Például a karakterkiejtés, karakterbeszúrás során a módosítási helytől a bekezdés végéig megtörténik a sormaradék igazítása, a karakterek esetleg több képernyősoron keresztül is elmozdulnak. Ha a beírás vagy beszúrás során a képernyősor betelik, egy új, üres képernyősor adódik hozzá a bekezdéshez. Ha a következő képernyősorban már egy másik bekezdés található, akkor az editor innen kezdve minden sort eggyel lejjebb léptet, beszúr egy üres sort, majd ezt hozzáfűzi a bekezdéshez. Ebből azt a tanulságot kell kiemelni, hogy az editor soha nem fűz össze két bekezdést!
Ha a karakterek kiejtése, vagy a bekezdés végének elhagyása, törlése során a képernyősor utolsó karakterpozíciója felszabadul, akkor az editor az ezt követő megüresedett sort a bekezdés végéről kitörli, az alatta levő sorokat pedig eggyel feljebb lépteti.
Ha az editornak a képernyő nemlétező 25. sorába kéne írnia, akkor működésbe lép a „roll" funkció, azaz a kép fölfelé görgetése: minden sor eggyel följebb lép, a 24. sor pedig üressé válik. Ez történik például akkor is, ha a kép utolsó karakterpozíciójába írunk.
Fontosnak tartjuk megemlíteni, hogy a kurzor lefelé mozgatásával nem lehet a roll funkciót kiváltani, valamint az is lényeges, hogy a roll funkciót nem lehet kikapcsolni.
Az editor input alatt tetszőleges szerkesztési műveletek elvégezhetők a képernyőn, amíg a bevitelt a RETURN billentyűvel le nem zártuk. Az aktuális input sor az a bekezdés lesz, amelyikben a kurzor a lezáráskor volt. A kép-

ernyőre egyszer kiírt karakterek – akár input, akár output írta oda – többször is beolvashatók, akár kis módosításokkal is.
Az outputtal kiírt vezérlőkarakterek általában olyan hatást váltanak ki, mint az input során használt megfelelő vezérlőbillentyűk. Három kivétel van:

1. 10 = 0Ah (LINE FEED, CTR + J)

Csak outputnál van hatása. A kurzort a bekezdést követő sorba viszi, míg az oszloppozíciót változatlanul hagyja. Szükség esetén a roll funkciót is végrehajtja.

2. 13 = 0Dh (RETURN)

Input esetén a bevitel lezárását jelenti. Outputnál a kurzort a sor első pozíciójába viszi, így az oszloppozíció 1 lesz, a sorpozíció változatlan.

3. 27 = 1Bh (ESC)

Csak inputnál van hatása: a hívó programnak egyből átadja az ESC kódját. Ha a CTRL billentyűvel együtt nyomjuk le, akkor a „STOP" hibajelzést is beállítja.
A többi vezérlőkarakter input és output esetén is azonos hatású:

4=04h CTRL+D, RIGHT	kurzor jobbra
5=05h CTRL+E, UP	kurzor föl
7=07h CTRL+G, SHIFT+DEL	a kurzorpozícióban levő karakter kiejtése
8=08h CTRL+H, DEL	a kurzortól balra levő karakter kiejtése
9=09h CTRL+I	a kurzor a következő TAB pozícióba: 1, 9, 17, 25, 33, 41, 49, 57
11=0Bh CTRL+K	törlés a bekezdés végéig
14=0Eh CTRL+N	új bekezdés beszúrása
19=13h CTRL+S, LEFT	kurzor balra
22=16h CTRL+V, INS	karakter beszúrás
24=18h CTRL+X, DOWN	kurzor le

AZ EDITOR RENDSZERVÁLTOZÓI

ASCII–MAP 1600 byte, címe 256=100h
A képernyőre írt karakterek ASCII kódját tárolja. Egy-egy sorhoz 64 byte tartozik.
$INK 1 byte, címe 2893=0B4Dh
A színfelbontástól függően az alsó egy, kettő vagy négy bit a tintaszínt (palettakódot) adja.
Közös változó a video rutinnal.
$PAPER 1 byte, címe 2894=0B4Eh
Mint a $INK, csak a háttér (papír) színére. Közös változó a video rutinnal.
LOCK KEY 1 byte, címe 2918=0B66h
Az aktuális kurzor (LOCK) állapotát mutatja:
0 normál karakterek (LOCK, normál kurzor)
1 nagybetűk (LOCK+CTRL, inverz C kurzor)
2 folyamatos shift (LOCK+SHIFT, inverz S kurzor)
8 alternatív karakterek (LOCK+ALT, inverz A kurzor)
Közös változó a billentyűzet rutinnal.

Cseh Tibor

SZOFTVER ÖTLETEK



KISBETŰK A GRAFIKUS KÉPERNYŐN C16, PLUS/4

Bizonyára sokan bosszúsan tapasztalták már, hogy a grafikus képernyőn nem lehet kisbetűket használni. Erre van egy viszonylag egyszerű megoldás. A 740-es címen ($02E4) található annak a címnek a felső byte-ja, ahonnan a CHAR utasítás a karaktereket olvassa, ha a grafikus képernyőre ír. Ez alapesetben $DØ, mivel a karaktertár kezdőcíme $D000. A probléma azért van, mert ha a kisbetűs üzemmódba váltunk, az interpreter nem írja át ezt a byte-ot $D4-re, vagyis a kisbetűket tartalmazó karakterkészlet címének felső byte-jára. Ez a módszer akkor is működik, ha új karakterek használata esetén a grafikus képernyőn akarunk írni velük. Ilyenkor is az új karakterkészlet felső byte-ját kell POKE-kal a 740-es címre beírni.

***Grósz Attila,* Békéscsaba**

ATARI 600/800 XL UNIVERZÁLIS MAGNÓ RUTIN

Az Atari gépeknél kényelmetlenséget okoz a tárterület egyszerű magnóramentése.

Az egyébként is nehézkes szekvenciális file minden egyes blokk kivitele után hosszú szünetet tart.

Az alábbi rövid szubrutin segít a problémán. Egy rövid gépi kódú rutin a magja, mely az operációs rendszer CIOMAIN rutinját használja fel. Alkalmas tetszőleges címtől kezdve tetszőleges hosszúságú memóriarész magnóramentésére és beolvasására a BASIC programok CSAVE-lésének megfelelő sebességgel. Gyári programok fejrészét is beolvashatjuk vele vizsgálat céljából.

A programot bebillentyűzés után LIST "C:" parancs segítségével mentsük kazettára.

Így ENTER "C:" paranccsal bármilyen programhoz csatolhatjuk, csak ügyeljünk arra, hogy ne legyenek azonos sorszámok. A 10–15 sorokat csak egyszer kell végrehajtani, ezek a főprogramhoz kerüljenek.

A szubrutint GOSUB 10 000 utasítással hívjuk meg. Előtte adjunk értéket a paramétereknek a következőképpen:

MOD = 4 : olvasáskor
MOD = 8 : íráskor
MS : kezdőcím
ML : hossz

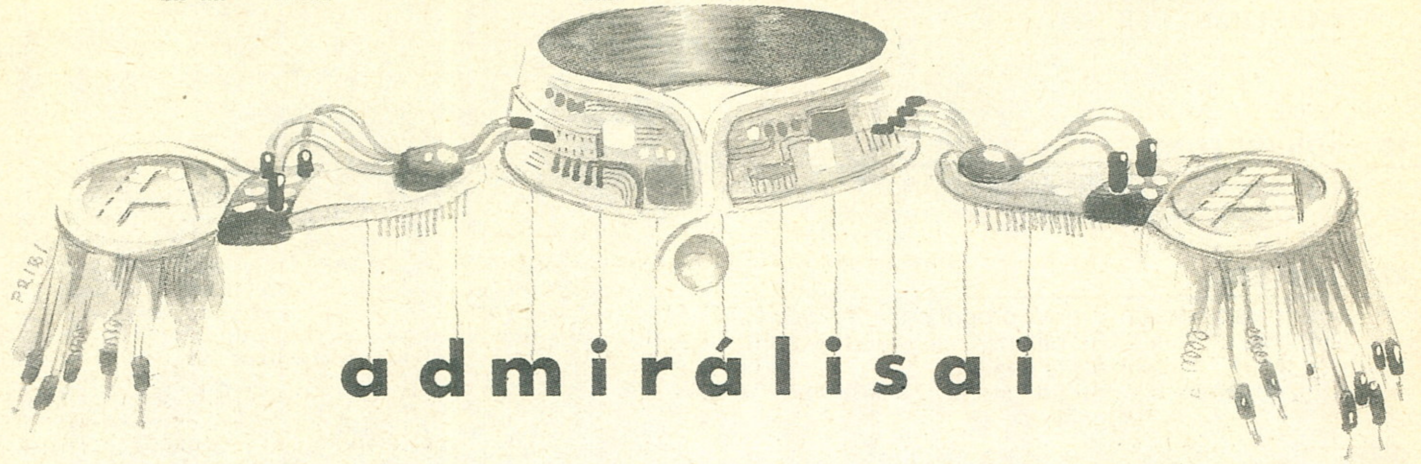
Remélem, sikerült valamivel kényelmesebbé tennem a gép használatát.

Brosig János, Szombathely, Bokai u. 13/A

```
10 DIM C$(30):C$(28,29)=" "
11 DATA 162,48,104,104,104,157,66,3
12 DATA 104,157,69,3,104,157,68,3,104
13 DATA 157,73,3,104,157,72,3,32
14 DATA 86,228,96
15 FOR I=0 TO 27:READ X:POKE ADR(C$)+I,X:
NEXT I
10000 REM MOD MS ML
10020 POKE 764,12
10030 OPEN#3,MOD,128,"C:"
10040 X=USR(ADR(C$).,MOD+1,MS,ML)
10050 CLOSE#3
10060 RETURN
```

A szerkesztő azért van, hogy a lap olyan legyen, amilyenek az olvasói!

AZ ADMIRÁL admirálisai



A Mikro '87 kiállításon nagy sikerrel mutatkozott be a FORTH programnyelvet ismerő ADMIRAL '64 típusú számítógép, amelyet Urbán Zoltán, a budapesti Landler Jenő Szakközépiskola érettségiző diákja és unokabátyja – az iskola tanára –, Urbán Kornél épített.

– Az iskola Computer Klubja már néhány hónappal a megalakulás után átlépte saját árnyékát – emlékezik a kezdetekre Urbán Kornél. **Nem tudtuk nyomon követni, hogy éppen ki mivel foglalkozik. Így 1986 közepén elhatároztuk, hogy minden fejlesztőmunkánknak önálló munkaszámot adunk. Ezek közül az első egy 2K EPROM-ot és 1K RAM-ot tartalmazó rendszer volt. A 02 sorszámtól viszont már kezdődnek a sikeres ADMIRAL-gép előzményei.**

Urbán Zoli veszi át a szót:

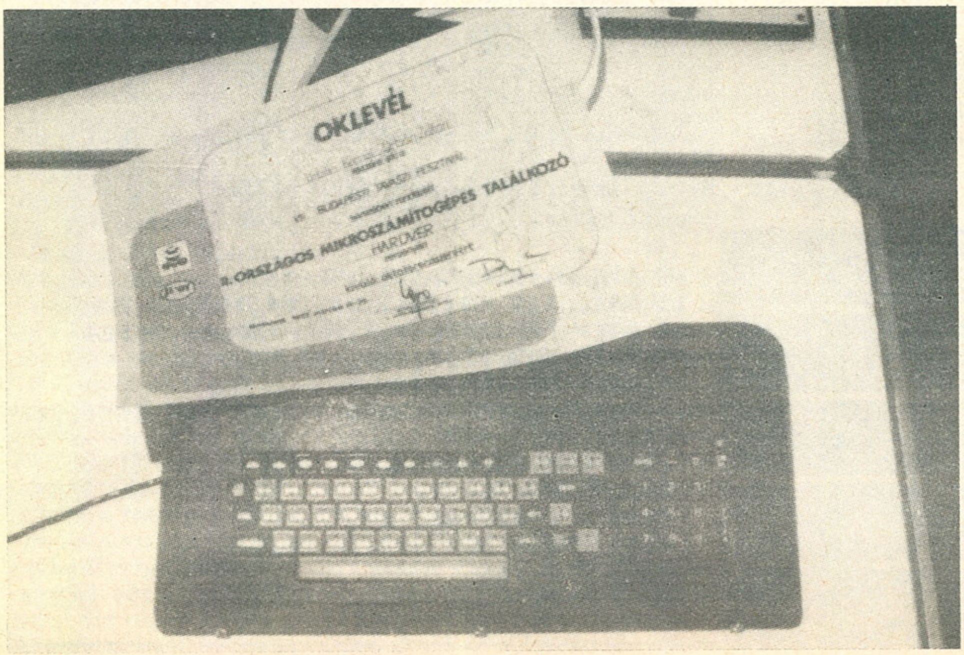
– Másodikos voltam, amikor a Mikroszámítógép Magazinban először olvastam a FORTH nyelvről. Megtetszett a kezelhetősége, hajlékonysága, gyűjteni kezdtem az ezzel kapcsolatos irodalmat, és nekiláttam egy FORTH-ban programozható gép megépítésének. Nagybátyám is lelkesedett, így az iskolában, a klub keretében is hozzáláttunk egy hasonló gép kidolgozásához. Közben 1985. végére otthon elkészült a saját gépem – ez persze „kisipari" kivitel volt, az áramköri lapokat még kézzel rajzoltam, és doboza sem volt –, de jól működött.

– A 02 számú rajzok – mondja Urbán Kornél – **már a gép itt készült változtatásának terveihez tartoznak. Eredetileg modulrendszerben akartuk megvalósítani, sőt el is kezdtük az elkészítést. Azután kiderült, hogy ez több ok miatt kivitelezhetetlen: a modulok közötti csatlakozók nagyon drágák lettek volna, és az érintkezési hibákat sem tudtuk kiküszöbölni. Elkezdtünk elveszni a részletekben, míg azután rá kellett jönnünk, hogy a gépet csak egységes egészként lehet megvalósítani.**

Később azután a fejlesztési munkákat e gépnek rendelték alá: a 03. terv tartalmazta az áramkör CPU kártyáját, a 04. az interface-kártyát, a 05. pedig az EPROM-égetőt.

– Az EPROM-égető gyártására először kerestünk vállalkozót, de végül is egyik ajánlattal sem voltunk elégedettek. Emiatt döntöttünk úgy, hogy inkább önmagunk gyártjuk le.

Az iskola jó partnerkapcsolatot tart fenn több vállalattal – így a Mikroelektronikai Vállalattal, az EMG-vel, és a Fotoelektronik-Novotrade GT-vel –, ők segítenek az alkatrészellátásban. Bizonyára ők is belátták, hogy ez a szakma egyre inkább elméletigényessé válik: olyan technikusokra van szükség, akik értelmesen tudnak gondolkodni, önállóan képesek dolgozni. A támogató intézmények felismerték, hogy jobb technikusokat tudunk kibocsátani, ha ők is hozzájárulnak a korszerű feltételek megteremtéséhez.



– Ez a szoba kicsit a senki földje, független terület. – mondja Urbán Kornél, amikor belépünk a Computer Klub szobájába. Úgy tűnik, hogy a szakközépiskola rendjéhez képest itt nem kevés engedményt élveznek a diákok: a falakon feliratok, poszterek, látszólag ötletszerűen kitűzött számítógépes listák. A szétszórt rajzok, félkész nyomtatott áramkörök viszont jelzik, hogy ez nem közönséges diákklub: itt komoly alkotómunka folyik. Ebben a teremben találkozunk végre az iskolában épített Admirallal is. Külsőre nem sokban különbözik a megszokott, BASIC-ben programozható gépektől.

– Valóban nem, hiszen a doboza és a billentyűzete egy leselejtezett, Videoton gyártmányú VSD 47700 típusú terminálból származik, így külsőre semmi sem árulja el, hogy a doboz egy teljesen új gépet tartalmaz. Ez persze a gép építésekor is peremfeltételt jelentett: a nyomtatott áramköröket úgy kellett kialakítani, hogy azok beférjenek a dobozba.

– Mi ragadta meg önöket a FORTH programnyelvben?

– Két évvel ezelőtt, amikor először hallottunk, olvastunk erről a nyelvről, úgy nézett ki, hogy hatalmas jövője lesz. Elébe akartunk menni a jövőbeli fejlődésnek, a számítástechnikai oktatás holnapjának. Elvégre amikor nekikezdtünk ennek a munkának, szinte már mindenhol oktatták a BASIC nyelvet. Úgy gondoltuk, hogy a FORTH frissítést jelent majd. Később aztán persze kiderült, hogy nem volt ebben igazunk – de más részről nagyon is megérte a gép kidolgozása.

– A legtöbb iskolában a fiatalok alig érinthetik meg a számítógépeket. Ha arra kíváncsiak, hogy mit tartalmaz a belsejük, akkor a tanárok inkább szakkönyvekhez küldik őket, mert féltik a drágán vásárolt géppeket. A mi gépünk viszont olyan, amelyet nyugodtan lehet vizsgálgatni, lehet rajta kísérletezni, hiszen a legdrágább alkatrész ára sem több, mint 300–400 Ft.

Urbán Zolit kérdezzük: milyen programokat írt az Admiralra?

– Az fogott meg a FORTH-ban, hogy ez a nyelv nem dolgozik állandó utasításkészlettel, hanem folyamatosan bővíthető. Így ezzel a nyelvvel kapcsolatban megelégedtem azzal, hogy a gépet kidolgoztuk – a többi legyen a programozók feladata. Az általánosan elfogadott, standard utasításkészlet bent van a gépben, ezentúl minden a programozó ügyességén múlik.

Írtam azért persze egy-két programot is, így az EPROM-égető programját, azon kívül néhány játék- és demo-programot, valamint egy dissassemblert egy másik processzorhoz – szóval efféle „apróságokat" – mondja Zoli.



A nyurga 18 éves fiú 9. helyezést ért el az idei Országos Szakmai Tanulmányi Versenyen – így felvételi vizsga nélkül folytathatja tanulmányait a Budapesti Műszaki Egyetem Villamosmérnöki Karán. Jelenleg éppen egy új gépet fejleszt, amely a meglévő Admiral '64-re épül, de rendelkezni fog grafikus terminállal, és lemezegységet is lehet majd kapcsolni hozzá. További tervei közé tartozik egy CP/M operációs rendszert használó gép, hiszen ez alatt a rendszer alatt bármi futtatható.

Urbán Kornél elégedett lehet az iskola, illetve a klub eredményeivel.

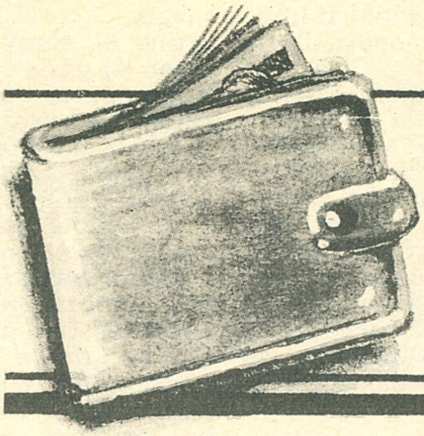
– Valóban az lennék, ha egy-két gond nem okozna álmatlan éjszakákat. Az egyik az, hogy a jelenlegi oktatói gárda már nem tudja sem létszám, sem szakmai felkészültség terén követni újonnan kialakult szakmai profilunk fejlődését. Hatalmas szükségünk lenne egy felsőfokú végzettségű fiatalra, aki tanári állást vállalna nálunk. A másik gondunk az, hogy mi ugyan megépítettük az Admiral 64'-et, de ezt szinte sehol máshol nem ismerik. Szeretnénk megteremteni az elterjesztés lehetőségeit, illetve szívesen átadnánk a gépünket – hiszen egy ilyen olcsó számítógép, mint ahogy már utaltam rá, szinte elronthatatlan, annyira olcsók az alkatrészei.

Tallér József

IBM kompatibilis gépek ügyében eszméletlen zűrzavar uralkodik e pillanatban idehaza. Pár hónapja megjelentek a híradások, miszerint vége a sötét üzelmeknek, mert a hazai szerelésű klónokra olyan központi támogatást kapnak bizonyos cégek, hogy attól azután pillanatok alatt lezuhan majd az ár. Vártuk, vártuk a zuhanást, ám közben gazdasági krízisünk folyamányaként a megígért dollárkeretek meghatározatlan időre „eltűntek". Így hát nemhogy zuhantak volna az árak, hanem még a gépeket is keresni kellett. Az árzuhanás kilátásai miatt a magánbehozatal is lefékeződött. Mindenki várakozó állásponton volt, s szokás szerint lábra kaptak a rémhírek, hogy sehol sem lehet gépet kapni, az árak erre szépen visszakúsztak a tavalyi szintre. Ebben a helyzetben érkezett el a BNV ideje. Mielőtt kimentünk, elhatároztuk, hogy IBM-ügyben alaposan tájékozódunk odakint. Úgy véltük – véljük –, hogy nagyon sok cég képviselője ment el a nemzetközi vásárra azzal, hogy szétnéz: tényleg mit lehet most kapni, mennyiért és milyen határidővel. Mi is egy ilyen tájékozódónak adtuk ki magunkat tehát. Nem árultuk el, hogy újságírók vagyunk. Egyszerűen csak odamentünk a standra és érdeklődtünk. Már előre halljuk egyes cégek tiltakozását: „ha meg akartak valamit jelentetni, miért nem közölték, miért nem keresték föl a sajtó- és propagandaosztályt?" Előre szeretnénk válaszolni: azért, mert úgy gondoljuk, hogy a sajtóosztályon keresztül kapott információknál sokkal fontosabb az, hogy milyen információkat kaphat az egyszerű vásárló, mondjuk a BNV-n. Úgy gondoljuk, hogy amelyik cég olyan embereket állított standjára, aki hamis, vagy hiányos információkat tudott csak adni, ez a cég megérdemli, hogy ezek az információk jelenjenek meg velük kapcsolatban.
Valószínűleg olyan cég is akad, amely be sem került táblázatunkba. Velük kapcsolatban is tiszta a lelkiismeretünk. A BNV számítógépes tájékoztató szolgálatától ugyanis megkértük az összes számítógépes cég listáját. Aki nem volt benne, arra vagy rábukkantunk sétánk során, vagy sem. Végül is tíz hazai PC forgalmazó kínálatából válogattunk. Válogattunk, mert voltak cégek, amelyek olyan sokféle konfigurációban hajlandók XT-t és AT-t is szállítani, hogy minden ajánlatukat nem akartuk közölni. Egyébként a legtöbb kiállító készségesen rendelkezésünkre bocsátotta legfrissebb árjegyzékét, de azért előfordult, hogy az árakat csak a hosszas keresgélés után előkerült illetékes szóban tudta velünk közölni. Volt, ahol nem sikerült választ szerezni olyan „lényegtelen" kérdésekre, hogy milyen interface tartozik a konfigurációhoz, vagy éppen azt nem tudtuk meg, hogy a vételár tartalmazza-e az installálás és a garancia költségeit. Volt, aki a szállítási határidő megjelölésénél jött zavarba, és közölte velünk, hogy mint minden hazai gyártó, ők is a harmadik vagy a negyedik negyedévben tudnak szállítani, egyes konfigurációkat pedig „majd". (Az ilyen „konkrétságú" ajánlatok nem szerepelnek a táblázatban.)

JELMAGYARÁZAT:

„mono"-val a monochrome alfanumerikus, „színes"-sel a színes grafikus video illesztőkártyát és monitort jelöltük.
„R" RS 232, „C" Centronics interface-t jelent.



M

Forgalmazó	Géptípus	Operatív tár/kbyte
Műszertechnika Kisszövetkezet	MXT I	256
	MXT II	256
	MAT III	512
	MT386	512
Data Manager Számítástechnikai Kisszövetkezet	IBM PC/XT	640
	IBM PC/XT	640
	PC/XT	640
	PC/XT	640
	IBM PC/AT	512
	IBM PC/AT	512
	PC/AT	512
	PC/AT	512
ECONORG Számítástechnikai Közös Vállalat	MULTITECH	
	POPULAR 500	512
	MULTITECH700	
	HGE TURBO	640
	MULTITECH	
	ACCEL 900	1 M
Csepel Művek Transzformátorgyár	Cs16XT turbó	256
	Cs16XT turbó	256
CONTROLL Kisszövetkezet	MC 86/A	256
	MC 86/W27	256
	MC 87/W	512
MICROSYSTEM Kisszövetkezet	P-XT	640
	P-XT	640
	P-AT	640
	P-AT	640
VIDEOTON Elektronikai Vállalat	VT 110	640
	VT 110	640
	VT 160	640
	VT 160	640
KSH SZÜV	IBM XT	640
	IBM XT	640
	IBM AT 02	512
	MEGAMICRO XT	640
	MEGAMICRO AT	1 M
SZKI	PROPER 16	512
SZÁMSZÖV Kisszövetkezet	XT	256
	AT	512

I MENNYI?

Floppy /byte	Winchester /Mbyte	Videokártya	Monitor	Interface	Ár ezer Ft	Installálás Ft	+Garancia hónap	Szállítási határidő
360 k	–	mono	mono	–	150	25000	12	2 hónap
360 k	27	színes	mono	R,C	320	45000	12	2 hónap
1,2 M	27	színes	színes	R,C	495	65000	12	2 hónap
1,2 M	27	színes	színes	R,C	950	120000	12	2 hónap
360 k	20	mono	mono	R,C	750			
360 k	20	színes	színes	R,C	790			
360 k	20	mono	mono	R,C	400			III.
360 k	20	színes	színes	R,C	440			
1,2 M	20	mono	mono	R,C	1300			
1,2 M	20	színes	színes	R,C	1350			negyedév
1,2 M	20	mono	mono	R,C	550			
1,2 M	20	színes	színes	R,C	590			
360 k	–	mono	mono	–	170	a vételár		III.
360 k	20	színes	mono	R,C	310			negyedév
						tartalmazza		
1,2 M	40	színes	színes	R,C	750			vége
360 k	27	mono	mono	R,C	367	a vételár		azonnal
360 k	27	színes	színes	R,C	390	tartalmazza		azonnal
360 k	–	színes	mono	R,C	157	23000	6	1 hónap
360 k	27	színes	mono	R,C	275	26000	6	1 hónap
1,2 M	27	színes	mono	R,C	480	41000	6	1 hónap
360 k	20	mono	mono	R,C	184	30000	6	1 hónap
360 k	20	színes	színes	R,C	234			1 hónap
1,2 M	20	mono	mono	R,C	238	42000	6	1 hónap
1,2 M	20	színes	színes	R,C	300			1 hónap
360 k	20	mono	mono		200			
360 k	20	színes	színes		254			6–9
1,2 M	20	mono	mono		260			hónap
1,2 M	20	színes	színes		314			
360 k	20	mono	mono		750			2 hónap
360 k	20	színes	színes		850			2 hónap
1,2 M	20	színes	színes		1350			2 hónap
360 k	20	mono	mono		350			azonnal
1,2 M	20	színes	színes		495			azonnal
720 k	20	mono	mono		290			azonnal
360 k	27	mono	mono		350	a vételár	nem	1–2
1,2 M	27	mono	mono		620	tartalmazza		hónap

a distron HIBÁI

A BIT-LET-ben taglalták a ZX Spectrumra írt Beta Basic program hibáit. Rucz Lajos – a második cikk szerzője – Rutinról rutinra című könyvében megemlíti, hogy a Distron című programnak (disassembler) is van hibája pl., hogy az RST utasítás után mindig hexadecimális számot ír (az említett mű 128. oldalán ír erről), a defb utasítást rosszul értelmezi stb. Arról viszont nem ír, hogy hogyan lehet kijavítani. Ha van rá lehetőség, kérjék fel a szerzőt egy rövid cikk megírására, melynek alapján a program tökéletesítését mindenki elvégezheti. Feltételezem, hogy nem csak én használom a DISTRON 48 programot, tehát nem feleslegesen foglalná a helyet a BIT-LET-ben egy ilyen jellegű cikk. Segítségüket előre is köszönöm.

***Hirth Tibor,* 6430 Bácsalmás, Szt. István u. 6.**

A „Rutinról rutinra" c. könyv (LSI ATSz. 1986) 128. oldalán valóban hivatkozás történt a Distron disassembler program egyik legfontosabb hibájára, vagyis arra, hogy decimális output (kijelzés) esetén is hexadecimálisan jelenik meg néhány gépi kódú utasítás, s ez gyakran zavaró lehet programjaink visszafejtése közben.

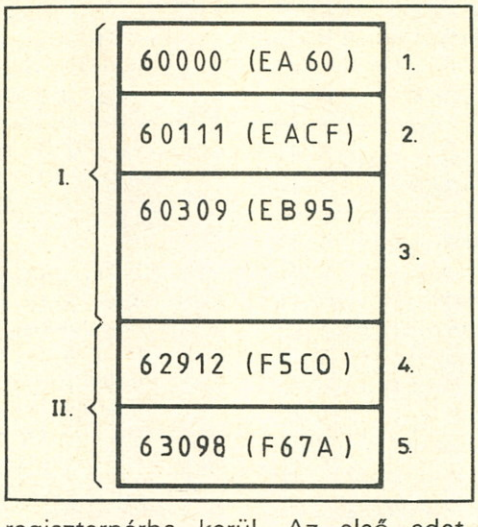
A hibakeresés és -javítás előtt nem árt, ha szólunk néhány szót a programról, ill. annak felépítéséről. A működésre részletesen nem térünk ki, hiszen ez megtalálható az említett könyvben.

A Distron előnyei:

- alkalmas decimális/hexadecimális disassemblált lista kijelzésére
- a disassemblált lista nyomtatóra küldhető
- az RST 08 és RST 40 (HEX RST 28) utáni disassemblált lista helyes („defb" szerkezetek)
- áthelyezhető a memóriában
- kezelése rugalmas, egyszerű

A Distron felépítése:

A Distront két, jól elkülöníthető részre bonthatjuk (I–II). Az I. rész maga a disassembler, a II. rész pedig az áthelyező mechanizmus. A disassembler belépési pontja a 60000 (HEX–EA60) cím. Az 1. blokkban történik a képernyő megnyitása, majd a DISTRON copyright és a menütábla bejelentkezése. Ezt követően három adat bekérését oldja meg a program – sorban a hexadecimális/decimális kijelzés beállítását, a disassemblálandó terület startcímét, végül a disassemblálandó terület végcímét. Eközben különböző szubrutinok kerülnek meghívásra, amelyet a 2-vel jelölt blokkban találhatók. Ezek a szubrutinok inicializálják a BASIC változóterületet, itt munkaterületet alakítanak ki (ezért törli a DISTRON – ha meghívjuk – a BASIC változókat). Itt történik a paramétervizsgálat, s a begépelt adat vagy adat kódja egy megfelelő regiszterbe ill.



regiszterpárba kerül. Az első adatbekérést (h/d=hexadecimális vagy decimális kijelzés) követően a 62897 (HEX F59F) címre 1 íródik be decimális esetben, ill. zérus, ha hexadecimális kijelzést kértünk. A második adatbekérést követően a 62873/74 (HEX F599A) memóriahelyekre kerül a startcím értéke, ill. a harmadik adatbekérést követően 62875K/76 (HEX F5A1/A2) memóriahelyekre íródik be a disassemblálandó terület végcíme. A 3. memóriaterület a disassembler fő végrehajtója. Gyakorlatilag ez is két jól elkülöníthető részből áll, de nincs értelme élesen elhatárolni. A 60309–61583 (HEX EB95–F0) terület a vezérlő rész. Itt történik a vizsgált memóriablokk visszáfejtése, a megfelelő kódokhoz a megfelelő mnemonikok hozzárendelése, majd a h/d választásnak megfelelően kijelzés a képernyőn. A 61584–62911 (HEX F091–F5BF) terület gyakorlatilag adattáblázat.

Első részében felhasználói adatmezőt találunk, majd a 61668 (HEX F0E4) címtől kezdődik a Z–80 mnemonikok tárolása stringként. Az első tárolt mnemonik a „nop", ezt követi az „ex af, af" és így tovább. Ha visszafejtjük a memóriát, azt tapasztalhatjuk, hogy az adatmezőben nem tisztán mutatkoznak a szövegkódok, ezt a program készítői a program egyszerűsítésére vezették be. A mnemonikok utolsó karakterének van ugyanis 128-cal nagyobb kódja.

Pl. a „nop" a következőképpen mutatkozik:

```
61668 (H-F0E4) - 110          "n"
61669 (H-F0E5) - 111          "o"
61670 (H-F0E6) - 240-128=112  "p"
```

A Distron második fő része (II.) az áthelyező mechanizmus. A 4. blokk az áthelyező program vezérlő része. Itt a 62945/46 (H–F5E1/E2), valamint a 63004/05 (H–61C/1D) címeken kell megadnunk az áthelyezni kívánt kód kezdőcímét, és a 63007/08 (H–F61F/20) címeken az áthelyezni kívánt kód hosszát. Eredetileg itt 60000 és 2912 lett beállítva. Az 5. blokkban helyezték el azoknak a memóriahelyzeteknek a címeit, amelyeket át kell számoltatnunk az áthelyezés után, hogy a program áthelyezett vál-

tozata is problémamentesen működjön.

A hibakeresés:

Tudjuk, hogy a program fő hibája az RST utáni számjegyek állandóan hexadecimális formában történő megjelenítése. Első lépés: megkeresni az RST stringet az adattáblázatban. Teszszük ezt azért, mert az RST az egyedüli utasítás a Z–80 gépi kódjában, melynek mnemonikja kétszámjegyű számot is tartalmaz. Feltételezhető, hogy a gyártó a táblázatba a hexadecimális számparamétereket helyezte el, s így a mnemonik megjelenítésekor ez mindig ebben a formában fog szerepelni. Más utasítás esetén azért nem probléma, mert az utasítás mellé tartozó kódérték a h/d választásnak megfelelően számítódik ki, ill. valós formában jelenik meg.

Ha végigpásztázzuk a memória megfelelő területét:

```
10 FOR i = 61584 TO 62911
20 LET a = PEEK i
30 IF a > 128 THEN LET a = a - 128
40 IF a < 32  THEN LET a = 32
50 PRINT i,PEEK i;"  ";CHR$ a
60 NEXT i
```

hamar felismerhetjük a következőt:

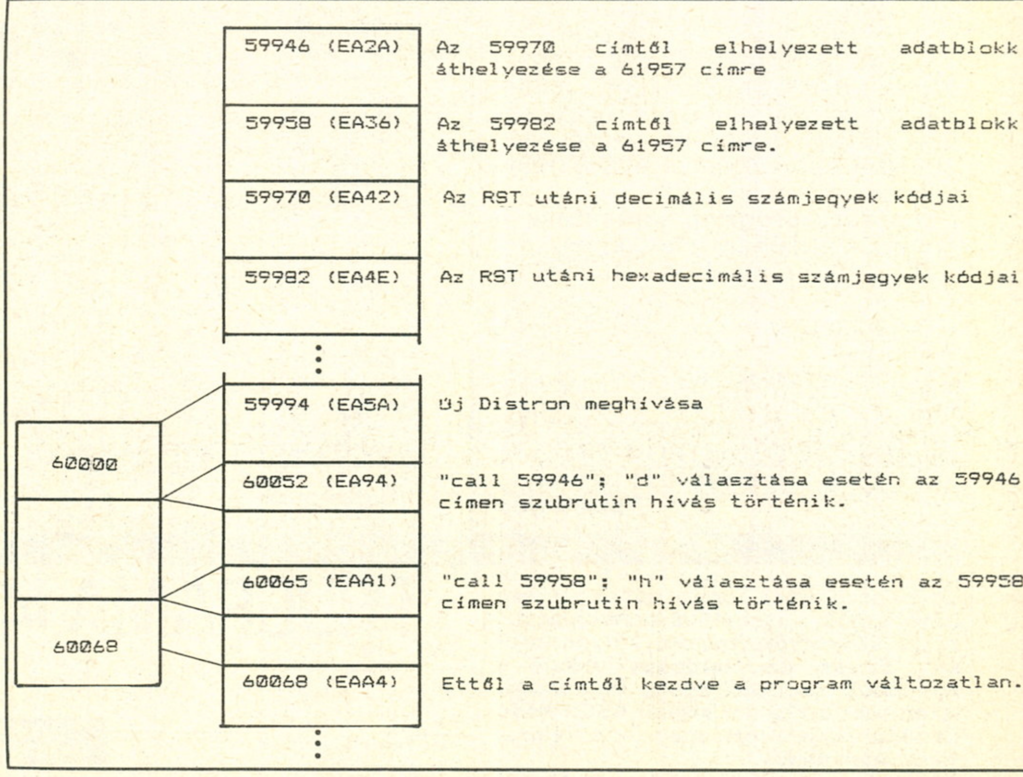
```
61947  114  r
61948  115  s
61949  244  t
61950  186  elválasztó kód
61951   48  0
61952  176  0 RST 00
61953   48  0
61954   56  8 RST 08
61955  163  elválasztó kód
61956   49  1
61957  176  0 RST 10
61958   49  1
61959  184  8 RST 18
61960   50  2
61961  176  0 RST 20
61962   50  2
61963   56  8 RST 28
61964  166  elválasztó kód
61965   51  3
61966  176  0 RST 30
61967   51  3
61968  184  8 RST 38
61969   58  elválasztó kód
```

Feladatunk egyértelmű, a h/d választásnak megfelelően az itt látható területet a megfelelő kóddal töltjük fel. **Térjünk vissza ismét a program 1. blokkjához.**

```
.
.
.
60032 17,112,244  LD   DE,62576
60035 1,31,0      LD   BC,31
60038 205,128,235 CALL 60288
60041 205,48,235  CALL 60208
60044 205,38,235  CALL 60198
60047 246,32      OR   32         ;ha "d"-t gépeltünk be,
60049 254,100     CP   100        ;akkor a 62879 címre
60051 32,7        JR   NZ,60060   ;1-et írunk, máskülönben
60053 62,1        LD   A,1        ;ugrás tovább
60055 50,159,245  LD   (62879),A
60058 24,8        JR   60068
60060 254,104     CP   104        ;"h" begépelése esetén
60062 32,4        JR   NZ,60068   ;a 62879 címre zérust
60064 175         XOR  A          ;írunk, máskülönben
60065 50,159,245  LD   (62879),A  ;ugrás tovább a 60068
60068 17,143,244  LD   DE,62607   ;címre
.
.
.
```

Itt rögtön a szemünk elé tárul a program egy másik hibája is, vagyis az, hogy a h/d választást követően akkor is tovább engedi a program futását, ha h-tól és d-től eltérő karaktert adtunk meg. Ilyenkor az éppen a memóriában levő értékek szerint fog történni a kijelzés. Ebből adódóan a 60062 címen levő ugró utasítást meg kell változtatni, jelen esetben a 60032 címre.

A helyes adatblokk elhelyezését csak úgy tudjuk megoldani, ha a h/d választás után lehetőséget biztosítunk egy kiugró utasítással az adott blokk átmozgatására.



Az adatblokk áthelyezését LDIR utasítással oldottuk meg. Az áthelyező mechanizmust és az adatokat itt külön nem részletezzük, azt az átjavítás után a Distronnal, saját magával megtekinthetjük. Egy fontos dolog: az új indító cím az 59994 (HEXEA5A) lett! Még egy probléma tisztázása van hátra, a Distron áthelyezhetőségét az új, javított változattal is lehetővé kell tennünk!

Az áthelyező mechanizmusban a megfelelő helyeken (ld. előbb) át kell írnunk az áthelyezendő blokk kezdőcímét 60000-ről 59946-ra, az áthelyezni kívánt blokk hosszát pedig 2912-ről 2966-ra. Az átszámításhoz tárolt címmezőt hátrább kell tolnunk annyiszor két byte-tal, ahány új abszolút címhivatkozást tartalmaz a program, majd ezeknek az értékeit (alsó/felső byte) el kell helyezni az adattábla elején.

Mindezek után nézzük meg az itt említett műveleteket végrehajtó BASIC segédprogramot és működését.

1. Töltsük be a DISTRON-t;

2. Gépeljük be a következő oldalon lévő BASIC programot:

A 10–60 sorok végzik el az áthelyező mechanizmus és az áthelyezéshez szükséges adatok elhelyezését a memória elején. A 70–100 sorok a DISTRON első részét módosítják, beszúrják a „call" utasításokat, átírják az ugró-hivatkozásokat.

A 110 sor az áthelyező mechanizmusban állítja be a kezdőcím és hossz értékét.

A 120–150 programsorok az áthelyezés után átszámítandó címek táblázatát hozzák rendbe.

A 160–210 sorok a bejelentkező címlapot írják át.

Az 1000–1020 programsorok segítségével generálhatunk egy olyan kódot az 50000–50010 memóriaterületen, amely bemutatja az összes RST

Bármely program bonyolultsága addig fokozódik, amig túl nem nő programozója képességein!

(Murphy törvénykönyve)

ötlet

a distron HIBAI

utasítás helyes működését a DISTRON-nal.

Ha a DISTRON a memóriában van, adjuk ki RUN, majd GOTO 1000 és ezután RANDOMIZE USR 59994. Válasszunk „d"-t, kezdőcímnek adjunk meg 50000-et, végcímnek 50010-et, és meglátjuk a DISTRON helyesen működik.

A javított program kimentése a következőképpen lehetséges: SAVE „DISTRON–2" CODE 59946, 3540.

A program továbbra is a RANDOMIZE USR 62912, majd az ezt követő cím megadásával helyezhető át. Figyelem! Áthelyezés után a DISTRON fog indulni! Az áthelyezett kód kimentése: SAVE „DISTRON–2" CODE xxxx, 2966 utasítással lehetséges.

Rucz Lajos

```
  10 FOR i=59946 TO 59993
  20 READ a
  30 POKE i,a
  40 NEXT i
  50 DATA 33,66,234,17,5,242,1,1
2,0,237,176,201,33,78,234,17,5,2
42,1,12,0,237,176,201
  60 DATA 182,50,180,51,178,52,4
8,166,52,184,53,182,176,49,184,5
0,176,50,56,166,51,176,51,184
  70 FOR i=60000 TO 60057: POKE
(i-6),PEEK i: NEXT i
  80 FOR i=60058 TO 60067: POKE
(i-3),PEEK i: NEXT i
  90 POKE 60052,205: POKE 60053,
42: POKE 60054,234: POKE 60065,2
05: POKE 60066,54: POKE 60067,23
4
 100 POKE 60046,10: POKE 60056,1
1: POKE 60060,221
 110 POKE 62945,42: POKE 62946,2
34: POKE 63004,42: POKE 63005,23

4: POKE 63007,150: POKE 63008,11
 120 FOR i=63473 TO 63112 STEP -
1
 130 POKE i+12,PEEK i: NEXT i
 140 FOR i=63098 TO 63123: READ
a: POKE i,a: NEXT i
 150 DATA 43,234,46,234,55,234,5
8,234,114,234,123,234,129,234,13
2,234,135,234,146,234,149,234,15
9,234,162,234
 160 FOR i=62693 TO 62705: READ
a: POKE i,a: NEXT i
 170 DATA 66,73,83,84,82,79,78,3
2,86,32,50,46,48
 180 FOR i=62741 TO 62747: READ
a: POKE i,a: NEXT i
 190 DATA 46,38,32,82,117,99,122
 200 FOR i=62755 TO 62769: READ
a: POKE i,a: NEXT i
 210 DATA 66,73,84,45,76,69,84,3
2,32,127,32,49,57,56,55
 220 STOP
1000 RESTORE 1020
1005 FOR i=50000 TO 50010
1010 READ a: POKE i,a: NEXT i
1020 DATA 199,207,55,215,223,231
,239,4,56,247,255
```

PROGRAMAJÁNLAT:

HT-1082ØZ + D100 GRAFIKUS HARDCOPY

Egyes programok komfortját jelentősen növelhetjük, ha lehetővé tesszük az értékes képernyőtartalmak kinyomtatását. Ez az ún. hardcopy-funkció a HT–1080Z gépeknél adott, ám a másolaton csak a 128-nál kisebb ASCII-kódú karakterek jelennek meg – a félgrafikus jelek nem.

Pedig ha erre a feladatra D-100-as nyomtatót használunk, a teljeskörű másolás is megoldható, hiszen a két eszköz grafikus karakterkészlete illeszkedik egymáshoz.

Erre mutat be egy igen egyszerű megoldást az alábbi program.

A működés megértésének segítésére számos megjegyzés került a listába – ezek természetesen elhagyhatók.

A 140-es sorig egy képernyőtartalom létrehozása zajlik. Hogy ez éppen ilyen lett, annak oka nem az öncélú gyönyörködtetés, hanem az, hogy alfanumerikus – grafikus jelek és szóközök mindenféle kombinációja előforduljon.

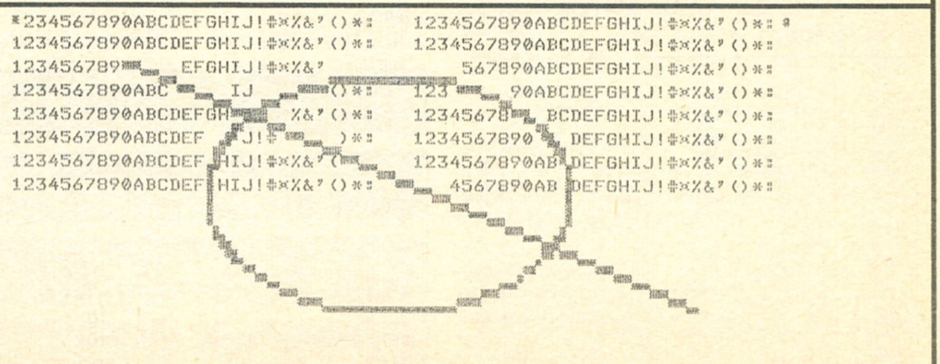
A lényegi rész a 190-es sorban kezdődő szubrutin.

A 200–220-as sorokban egy-egy képernyősor karaktereinek kiszedése és szétválogatása történik: a két string egyikébe az épp odavaló karakter, a másikba a megfelelő szóköz kerül. Ebből is sejthető, hogy a nyomtatás két menetben történik: először az alfanumerikus soré, majd a kocsi visszavitele után ugyanonnan a 2 félsorból összetevődő grafikus soré. A feltételek kiküszöbölik üres sorok „írása" esetén a kocsi felesleges mozgatását.

A 240-es sorra csak akkor van szükség, ha a nyomtatóhoz adott segédprogram nincs betöltve – egyébként a 270-es sorban a „kocsi vissza" után megtörténik a soremelés is. (Ha a segédprogram is be van töltve és a 240-es sor is benne marad a programban akkor printeléskor 1–1 üres sor lesz minden, kiprintelt sor után.)

A program BASIC ugyan, de a nyomtató miatt mégsem tűnik lassúnak. A CLEAR600 soknak tűnhet, ám így nem telik sok idő a stringműveleteknél a helykereséssel.

```
10 ' ***************************************************
20 ' **     HT 1080Z - D 100  Grafikus hardcopy      **
30 ' ***************************************************
40 ' ======== Timar Ferenc, Szentes, '8704. =========
50 '
60 CLS: CLEAR 300: DEFINT I-L
70 I$="1234567890": J$="ABCDEFGHIJ": K$="!#$%&'()*:": J=30
80 ' A kepernyo feltoltese (demo)
90 ' """"""""""""""""""""""""""""""
100 FOR I=0 TO 7: PRINT I$;J$;K$;"    ";I$;J$;K$: NEXT
110 FOR M=0 TO 6.28 STEP1/J: K=62-J*COS(M): L=23-J*SIN(M)/2: SET(K,L): NEXT
120 FORI=18 TO 112: SET(I,I/3): NEXT
130 SET(0,0): SET(0,47): SET(127,47): SET(127,0): ' Sarokjelek
140 PRINT@ 972, "H T  1 0 8 0 Z  -  D  1 0 0   H A R D C O P Y";
150 GOSUB 190
160 A$=INKY$: IF INKEY$="" THEN 160 ELSE CLS: LIST
170 ' Hardcopy-rutin
180 ' """"""""""""""""
190 K$=CHR$(128): K=253: L=15360: ' Gr.space, printer-port, kepernyo kezdet
200 FOR I=0 TO 15: I$="": J$="": FOR J=0 TO 63: L$=CHR$(PEEK(L+I*64+J))
210 IF L$<K$ THEN I$=I$+L$: J$=J$+CHR$(128) ELSE I$=I$+" ": J$=J$+L$
220 NEXT: ' A sor karaktereinek szetvalogatasa  (I$=alfan. J$=graf.)
230 I$=STRING$(6,32)+I$: J$=STRING$(6,128)+J$: ' Baloldali margo
240 OUT K,10: ' Soremeles (betoltott illesztoprogram eseten nem kell!)
250 IF I$<>STRING$(70,32) THEN LPRINT I$;: ' Alfanum. sor irasa
260 OUT K,13: OUT K,27: OUT K,54: ' Kocsi vissza, grafikusba valt
270 IF J$<>STRING$(70,128) THEN LPRINT J$ ELSE LPRINT K$: ' Graf. sor
280 OUT K,27: OUT K,53: NEXT: ' Alfanumerikusba valt, uj kepernyosor
290 FOR I=0 TO5: OUT K,10: NEXT: RETURN: ' 6 soremeles es vissza
```



H T 1 0 8 0 Z - D 1 0 0 H A R D C O P Y

SORVEZETŐ

RENDEZÉS

1. feladat: Keressük ki N szám közül a legnagyobbat (legkisebbet)!

```
10 INPUT N
15 DIM A(N)
20 FOR I=1 TO N:A(I)=RND(1)*100:PRINT
A(I),:NEXT I
100 MAX=A(1)
110 FOR I=2 TO N
120 IF A(I)>MA THEN MA=A(I)
130 NEXT I
150 PRINT:PRINT MA
```

A 20-as sort általában INPUT-tal használjuk (INPUT A(I)), de a program oktatásához az RND sokkal gyorsabb! A legkisebb elem keresésénél a 120-as sorban megfordul a relációjel.

F1. Írjuk ki, hányadik elem(ek) bizonyult(ak) legnagyobbnak! Két szám maximumát előállítja az alábbi aritmetikai kifejezés:

MAX(A,B)=(A+B+ABS(A—B))/2

MIN(A,B) =(A+B—ABS(A—B))/2

A fenti összefüggések gyakorlati jelentősége nem egyértelmű, csupán érdekességként említettük meg. Megjegyezzük, hogy amely gépen van DEF FN (X,Y) – például az új HT-n, azon így állna elő legegyszerűbben négy szám maximuma:

MAX(X, Y, U, V)=FNA(FNA(X,Y),FNA(U,V)),

ahol DEF FNA(A,B)=(A+B—ABS(A—B))/2

2. feladat: Rendezzünk N számot nagyság szerint!

Visszavezetjük a feladatot az előzőre:
10–20. sorok az előzőből, a többi:

```
10 INPUT N
15 DIM A(N)
20 FOR I=1 TO N:A(I)=RND(1)*100:PRINT
A(I),:NEXT I
30 PRINT
90 FOR J=1 TO N-1
100 MAX=A(J):K=J
110 FOR I=J+1 TO N
120 IF A(I)>MA THEN MA=A(I):K=I
130 NEXT I
150 PRINT MA,:A(K)=A(J):A(J)=MA
160 NEXT J
180 PRINT A(N)
```

A maximális elemet mindig a részsorozat elejére visszük (A(J)), és a továbbiak között újabb „legnagyobbat" keresünk. Lényegében ez történik az alábbi jóval egyszerűbb változatban (vö. Technika 1984/III. ill. Simonovits könyv 330–332. oldalak).

```
10 INPUT N
15 DIM A(N)
20 FOR I=1 TO N:A(I)=RND(1)*100:PRINT
A(I),:NEXT I
30 PRINT
100 FOR J=1 TO N-1
110 FOR I=J+1 TO N
120 IF A(I)>A(J)  HEN A=A(I):A(I)=A(J):A
(J)=A
130 NEXT I
140 PRINT A(J),:NEXT J
150 PRINT A(N)
```

Hangsúlyozzuk a tanulóknak, hogy egyetlen (N hosszú) FOR ciklussal nem lehet rendezni!

F2. Hasonlítsuk össze a különféle rendezéseket!

F3. Rendezzünk stringeket (neveket) névsorba! Figyeljünk külön a nem betű karakterekre!

F4. Tegyük fel, hogy minden névhez egyéb adatok is tartoznak (kor, osztályzat, fizetés, lakhely, stb). Gondoskodjunk arról, hogy a rendezett listában is a megfelelő adatok kerüljenek a nevek mellé!

F5. Rendezzük a neveket az egyes adatok szerint!

F6. Legyen N=10. Demonstráljuk a rendezést: a cseréket bonyolítsuk le a képernyőn. I-edik és J-edik elemek „vonuljanak" új helyükre, ha kell.

3. feladat: Tegyük fel, hogy van egy N elemű rendezett listánk. Szúrjunk be egy további elemet a listába, a megfelelő helyre!

a. megoldás. Eddigiek alapján könnyen adódik:

```
10 INPUT
"A RENDEZETT LISTA ELEMSZAMA=";N
15 DIM A(N+2)
20 FOR I=1 TO N:A(I)=I*I:PRINT
A(I),:NEXT I
50 INPUT "AZ UJ SZAM=";A
90 L=N+1
100 FOR I=1 TO N:IF A(I)>A THEN
L=I:I=N+2
110 NEXT I
200 A(N+1)=A(N):FOR I=N+1 TO L+1 STEP
-1:A(I)=A(I-1):NEXT I
210 A(L)=A:N=N+1
250 FOR I=1 TO N:PRINT A(I);:NEXT I
```

b. megoldás. Egy lényegesen gyorsabb változat:

```
10 INPUT
"A RENDEZETT LISTA ELEMSZAMA=";N
15 DIM A(N+2)
20 FOR I=1 TO N:A(I)=I*I:PRINT
A(I),:NEXT I
50 INPUT "AZ UJ SZAM=";A
90 K=0:L=N+1
100 M=INT((K+L)/2):IF L-K<1.5 THEN 200
110 IF A<A(M) THEN L=M: ELSE K=M
120 GO TO 100
200 A(N+1)=A(N):FOR I=N+1 TO L+1 STEP
-1:A(I)=A(I-1):NEXT I
210 A(L)=A:N=N+1
250 FOR I=1 TO N:PRINT A(I);:NEXT I
```

Figyeljük meg, N=100 esetben hány vizsgálatot (110. sor) végez a két program?

4. feladat. Tegyük fel, hogy van két sorbarendezett listánk /A(I) és B(J)/. Fésüljük össze ezeket egyetlen rendezett sorozattá!

Használjuk az előző feladat b. megoldásának ötletét!

```
10 INPUT "A KET RENDEZETT LISTA ELEMSZAM
A=";N,M
15 DIM A(N+2),B(M+2),C(N+M+2)
20 FOR I=1 TO N:A(I)=I*I:PRINT A(I),:NEX
T I
30 PRINT:FOR I=1 TO M:B(I)=I*I*I/4:PRINT
 B(I),:NEXT I
90 PRINT:I=1:J=1:K=1
100 IF A(I)>B(J) THEN C(K)=B(J):J=J+1:K=
K+1: ELSE C(K)=A(I):I=I+1:K=K+1
120 IF I<N+.5 AND J<M+.5 THEN 100
150 IF I>N AND J>M THEN 250
200 IF I>N THEN L=M-J:FOR X=0 TO L:C(K+X
)=B(J+X):NEXT X:GO TO 250
220 L=N-I:FOR X=0 TO L:C(K+X)=A(I+X):NEX
TX
250 FOR I=1 TO N+M:PRINT C(I);:NEXT I
```

Valami hasonlóra gondolhattak a „Tippek és trükkök C–64-re" c. DATA-BECKER könyv szerzői, például amikor a gyors rendezés programjukat kitalálták. A közölt BASIC program teljesen hibás, csak egészen speciális esetekben működik jól. Bizony, kiváló könyvekben is előfordulhat tévedés, sajnos kitalálni sem tudom, mire gondolhattak a szerzők, így a javítás is reménytelen. A rendezésekről egyébként sok érdekes és garantáltan igaz dolog olvasható Lovász–Gács: Algoritmusok. Műszaki Kiadó, 1981. könyvében. A 4. feladat adja azt az ötletet, hogy egy sorozatot a következőképpen rendezzünk: két részre bontjuk, és ezeket külön rendezzük, majd a két részt összefésüljük. Természetesen a két részt is tovább bontogatjuk, amíg csak lehet. Így az eredeti feladatot elemek cseréjére és összefésülésekre bontottuk.

F7. Rendezzünk 128 elemet sorba ily módon!

Török Turul



KÖNYVMOLY

Pethő Ádám: Assembly alapismeretek – SZÁMALK, 139. Ft.
(A háromrészesre tervezett sorozat első része nyolc fejezetben ismerteti az IBM PC/XT programozásához legszükségesebb tudnivalókat.)

Pyle: Az ADA programozási nyelv – Műszaki Könyvkiadó, 310. o., 108 Ft.
(A főleg rendszerek irányítására, felügyeletére alkalmas programnyelv részletes, referenciaszintű leírása.)

Bartel–Kraas–Schrüfer: Számítógép és sakk – DATA BECKER – Novotrade, 335 o., 390 Ft.
(A kötet a számítógépes sakkprogramok elméletébe vezet be, de az olvasó hasznos ismereteket szerezhet valamennyi stratégiai jellegű játék programozásával kapcsolatban.)

Dullin–Strassenburg: Az EPSON nyomtatók könyve – DATA BECKER – Novotrade, 256 o., 290. Ft.
(A gépkönyveknél jóval részletesebb mű kitér a táblázat- és formátumvezérlésre, a grafikus üzemmódokra, valamint a nyomtató belső felépítésére is.)

Neumann János és a „magyar titok" – OMIKK, 239 o., 114 Ft.
(A jól szerkesztett mű Neumann János eddig ismeretlen dokumentumaiba – önéletrajz, iskolai bizonyítvány, levelezés – enged betekintést.)

Első könyvem a mikrókról – Műszaki Könyvkiadó – Novotrade, 47 o., 99 Ft.
Szép, színes, gazdagon illusztrált könyvecskét kap kezébe az olvasó, ha megveszi a két magyar kiadó által közösen megjelentetett kötetet. Első pillantásra úgy tűnik, mintha tényleg minden igényt kielégítene, mintha bármelyik gyerek kezébe odaadhatnánk: „Nézd át, olvasgasd, ebből a képeskönyvből választ kapsz minden kérdésedre!" Csakhogy... Már a címadás körül is gondok vannak. Magyarországon legjobb tudomásom szerint a kutya sem emlegeti „mikró"-ként az otthon használatos mikroszámítógépeket. De ez lenne a legkisebb hiba, ha a kötet érdemi része megfelelne várakozásainknak. A legfőbb baj az, hogy a kötet szerzői – és főképp a magyar kiadók – egyáltalán nem döntötték el, hogy a számítógéppel ismerkedő gyerekek melyik csoportjának szánják kötetüket. Azoknak vajon, akiknek otthonában már amúgy is van számítógép, csak eddig kizárólag a papa (mama) használta; azoknak esetleg, akik szakkörbe járva tanulgatnak programozni, vagy pedig azoknak, akik egyik csoportba sem tartoznak: csak messziről láttak gépet, de nem idegenkednek tőle, sőt szeretnék tudni, hogy mit is kell csinálniuk, ha billentyűzet-közelbe kerülnek, s hogy egyáltalán mire is jó a masina.
Az angol szerzők álláspontja mintha egyértelmű volna: ők talán az utóbbi csoportot célozták meg – elvégre Angliában bizonyára van olyan jó anyagi helyzetű csemete, aki úgy jut számítógéphez, hogy sem neki, sem szüleinek fogalma sincs annak kezeléséről, használatáról. Nálunk azonban nem ez a helyzet: a gépet általában nem ilyen alapszintű könyvből tanuljuk. Ebből ered azután, hogy a magyar olvasó számára sok minden nehezen érthető a kötetben.
Így már az első fejezetben arra hivatkoznak a szerzők, hogy ha erről meg arról többet szeretnénk megtudni, akkor olvassuk el a gépkönyv megfelelő fejezetét. Nos, kíváncsi lennék, hogy a magyar kiadás nyolc-tíz-tizenkét éves olvasói közül ki az, aki az általában rosszul fordított (sőt gyakran lefordíthatatlan) gépkönyvekből ki tudja hámozni az őt érdeklő részeket. Képeskönyvünk rengeteg olyan szakkifejezéssel is szolgál, melyeknek megértése az angol anyanyelvű olvasóktól talán elvárható, a magyaroktól azonban bizonyosan nem. Elvégre a szerzők tegezve és közvetlen hangon szólítják meg az olvasót, így nyilván a kiskamaszokhoz szólnak – de mit kezd a számítástechnikában teljesen járatlan magyar kissrác az input, output szavakkal? A hasonló kifejezések – interpreter és társai – hemzsegnek a könyv folytatásában is.
Hasznos, persze a gép üzembe helyezéséről szóló fejezet – csak azt nem tudni, hogy ki számára. Ugyanis itt ismét visszakanyarodhatunk a legelőször feltett kérdésre: Magyarországon kevés az olyan gyerek, aki mindenfajta szülői tanács nélkül, pusztán e könyv alapján önállóan akarna dolgozni gépével.
A programozás alapjaiba bevezető részek csak elveszik a gyerekek kedvét a géppel való komolyabb ismerkedéstől. Mintaként két programlistát mutatnak be, ezek közül már az első is – tizennégy soros – egy FOR–NEXT ciklust, és egy IF–THEN elágazást tartalmaz. A magyarázat mindehhez alig egy-két szó. A számrendszerek, valamint a logikai kapuk működésének leírása pedig a matematikában jártasabb olvasót is fejtörésre készteti.
A könyv befejező részében azután engednek a szerzők a csak otthoni felhasználóknak szánt gépismertetésből. Engednek abból a követelményből, hogy ha törik, ha szakad, a gyereket meg kell tanítani a gép kezelésére. S ez az engedmény a kötetnek is hasznára válik.
Szólnak itt a chipek működéséről, a számítógépek történetéről – mindenki számára jól megérthető, használható módon. Érdekes alkalmazási példákat is megemlítenek – ezek közül a magyar olvasó számára nem egy még utópiának tűnhet –, de mindenesetre reménykedjünk abban, hogy a többrészesre tervezett sorozat ebben a szellemben folytatódik tovább.

Tallér József



P O S T A



A márciusi számunkban megjelent Primo stopper óriás számjegyekkel című anyagunkhoz kívánja a szerző *Fekete György* kiegészíteni a következőket:

1. A 3180-as sor teljesen a következőképpen néz ki:
3180 J=1: K=9: L=18: M=26:R=16443

2. Egész képernyő nagyságú számjegyekkel is képes az óránk dolgozni. Ehhez a következő változtatást kell elvégezni a programban:
1. Törölni kell a 130, 1010, 3020-as sorokat.
2. Át kell írni az alábbiakat:

```
190 NEXT: NEXT: NEXT: NEXT
230 IF Z="0" THEN GOSUB 1020: GOTO 110
3170 POKE F,33,0,A,62,5,245,229,14,7,26,21
3,17,31,0,6,27,119,35,119,25,16,250,209,19
,13,32,238,225,35,35,241,61,32,227,201: RE
M szamjegyrajzolo
3180 K=0: L=11: M=22: R=16443
3200 POKE T,1: CLS: PRINT$ 0,13,Z;: PRINT$
 15,13,Z;: GOSUB 1020: RETURN
```

Április számunkban tévesen közöltük Solti András lakcímét. Szerzőnk, aki a C 64-re írt életjáték programot adta közre, érthetően nem örült, hogy „Budapestre költöztettük". Így hát elnézését kérjük, s ezennel tudatjuk, hogy továbbra is Hódmezővásárhelyen él.

Szerkesztőségünk fontosnak tartja értesíteni a cikket küldő olvasókat, hogy a továbbiakban a következőknek meg nem felelő cikkjavaslatokkal nem foglalkozunk:

1. A cikk leíró része **olvasható kézírással** vagy írógéppel tetszőleges méretben, vagy szövegszerkesztővel **ékezetekkel**, vagy anélkül, esetleg **rovásírással** van írva
2. Névvel, címmel ellátott írásokat kérünk, ha a személyi szám is ott van, gyorsabban tudunk fizetni a megjelent anyagokért.
3. A mellékelt listát (listákat) ha van kérjük, **ha nincs, az sem baj.** Hogy jól **olvasható-e**, és hogy hány karakter van egy sorban, az azután igazán **édesmindegy**, hiszen a listakészítés alapvetően a szerkesztőség dolga.
4. **Az ábrák** – ha vannak – **lehetnek randák is**, ha közérthetetlenségek is vannak benne, akkor a legjobb ha a szerző mindjárt azt is megírja, hogy mikor és hol válaszol a felmerülő kérdésekre.

Összegzésként: mindegy, hogy hogy küldik az anyagot, csak zseniális legyen! A többit majd csak elegyengetjük valahogyan!

A következő levélre könyvnyi válaszokat kellene adni. Ezért a levélíró (*Bátor Ferenc*), 1211 Bp., Posztógyár u. 10.) figyelmébe ajánljuk mindenekelőtt a TVC Klubot, amely a Marcibányi téri Művelődési Házban működik.
Íme a levél, s nyúlfarknyi válaszaink.

A TV Computer gépi kódú programozása közben rengeteg kérdés vetődik fel. Íme belőle egy kis ízelítő:

1. Honnan lehet lekérdezni a botkormány elmozdulásának megfelelő kódokat? (Itt nem a 2905-ös címre gondolok).
Be lehet olvasni karakterként (GET vagy INKEY)

2. Át lehet-e szervezni a Video-memóriát, és hogyan? (Milyen szépen lehetne grafikázni négyszínű üzemmód helyett 16 színnel – persze kétszer akkora memóriával.)
Nem.

3. Melyik portról lehet beolvasni kazettáról?
Port 59H, 5-ös bit = magnó adat input

4. Meg lehet-e szerezni a RAM-bővítés kapcsolási rajzát?
(Talán meg tudom csinálni.)
Nyomdában van a Hardver kézikönyv, abban lesz sok hasznos információ. Kb. 2 hónap múlva jelenik meg.

5. Gépi kódú programot hogyan lehet (kiküldeni és beolvasni) kazettára, ha nem a BASIC tartalmazza pl. egy REM-sorban?
Nyomdában van az Operációs rendszer kézikönyv, abban minden benne lesz. Talán mire e sorok napvilágot látnak már kapni is lehet.

A 6., 7. kérdésre is ezek a könyvek adnak választ.

8. Megírtam hozzá egy egyszerű SRROLL-t. A program egyszer végez blokkmásolást, mégis lassú. Volt szerencsém látni a Szánkóversenyt, és itt elég gyorsnak tűnt. Ezt hogyan érték el? Átírogatták a Video kezdőcímet? Úgy tippelem, hogy a gép – míg az én programom fut – mást is csinál, amit nem várok el tőle. Hogyan lehet letiltani a felesleges munkáját? Vagy tényleg ilyen lassú lenne?
a) A Szánkóversenyt a Novotrade írta, tőlük kell megkérdezni.
b) Le kell tiltani az iterruptot, akkor biztos nem fog mást csinálni.

9. Milyen rendszer alapján végezték a gép formatervezését és a botkormány szerkesztését? (Én pl. házilag lehet, hogy jobb botkormányt tudok csinálni.)
Tudomásunk szerint a botkormányok közül nem sok romlott el. Ha az öné mégis, vigye be a szervízbe, meg fogják csinálni. Ha tényleg olyan jó botkormányt tud csinálni, a BIT-LET szívesen leközli a hardver leírását.

10. A BNV-n láttam egy Enterprise nevű számítógépet, és furcsán ismerősnek tűnt. Milyen gyártmány?
Angol gép. A hasonlóság többek közt abból ered, hogy a két gép BASIC-jét ugyanaz az angol cég fejlesztette.

VAN EGY ÖTLETE?

Kérdeztük februári számunk pályázatában. Sajnos olvasóink „NINCS"-csel válaszoltak, mert összesen 10 megoldást érkezett, s azok is inkább szegények voltak használható ötletekben, mint gazdagok. Az ígért fődíjat, az 1 doboz disket azért kiadjuk, de az összes többi díjat visszatartjuk, s most meghirdetett egy hónapos pályázatunkra ismét kitűzzük azokat.

Nos; a legjobb pályázatot egy katonaságnál alakult számítástechnikai szakkör tagjaitól kaptuk. Ők, a **ceglédi MN 1319 KISZ-Bizottság szakkörének** tagjai kapják meg a doboz disket. Íme, néhány ötletük. Zárójelben jelezzük, hogy melyek megírására keresünk vállalkozókat. Kérjük, hogy akinek ideje és kedve engedi, az jelentkezzen szerkesztőségünkben! Természetesen a megvalósult és használható programokat lapunkban közöljük majd.

Hardver nyomkövető – Képernyőn a számítógép tömbvázlata (regiszterek, RAM, ROM, DATA-, ADDRESS busz), s a gépi kódú utasítások vagy programrészek futása közben a hardver egységek működését mutatja.

Primo magyar hibaüzenetek (megírására vállalkozót keresünk!)

Több képernyő kezelése – (Videotonra, Plus/4-re megírására vállalkozót keresünk.)

Spectrum Full Screen Editor!! – (állítólag ők maguk megírják!)

Morze-fonia – a gép által (bemeneten) vett morzejelek kiértékelése és beszédszintetizátorral élő beszéddé alakítása. (hmm ...)

Folyamatábra – a gépbe betöltött program folyamatábrájának megjelenítése a képernyőn (az ötletadók vállalták a megírását Spectrumra. Kíváncsian várjuk az eredményt, de szívesen vesszük, ha más gépre akad vállalkozó!)

A további pályázók közül egy féléves BIT-LET előfizetéssel jutalmazzuk **Szuhár Gabriellát** és **Róka Sándort**. Az ő és más pályázóink beküldött ötleteiből legközelebbi számunkban közöljük majd az érdekesebbeket.

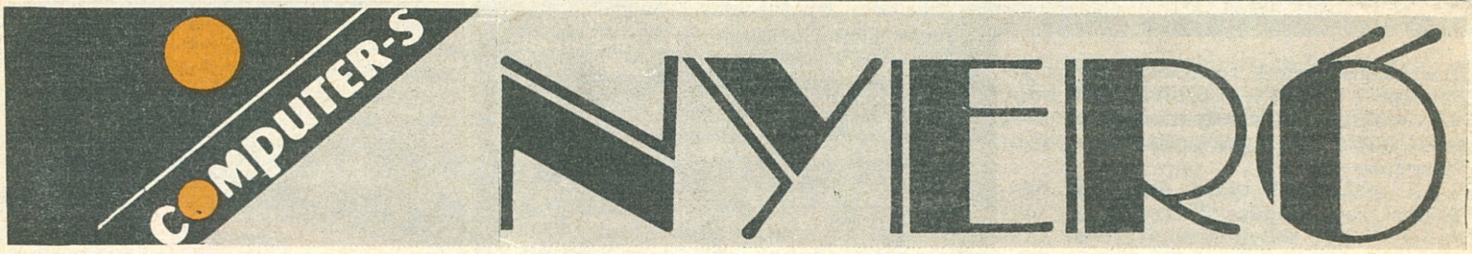
A PLUS/4 NYERŐ ÉRTÉKÉRE

A feladatok elég nehéznek bizonyultak, különösen a 2. feladat (programból stratégia visszafejtése), máskor ilyen feladatot nem adunk fel. Így összesen 19 pályázó küldött be mindhárom feladatra megoldást, s ebből 10 pályázónak csak 1 (vagy egy se)! teljesen jó megoldása volt. Így a gépet a legalább 2 jó és 1 részben jó megoldást beküldők között sorsolják ki, a megszokott súlyozott sorsolással. 1987. július 18-án délelőtt 11 órakor lesz az Almássy téri Szabadidő Központ Compánia számítógépes műhelyében.

Kozma Ágnes,	Miskolc – 3 cédulával
Kaczur István,	Győr – 3 cédulával
Koszper Vilmos,	Budapest – 3 cédulával
Kurusa Árpád,	Szeged – 2 cédulával
Lugosi Antalné,	Budapest – 2 cédulával
Puskás Pál,	Szigethalom – 2 cédulával
Biczó Anikó,	Sopron – 1 cédulával
Holbok Ferenc,	Tét – 1 cédulával
Nyéki Péter,	Pápa – 1 cédulával

AZ ATARI NYERŐ 2. FELADATÁNAK MEGOLDÁSA

```
10 INPUT H,N
20 PRINT MID$("SZERDA   CSUTORTOKPENTEK
  SZOMBAT  VASARNAP HETFO    KEDD     ",
((H-1)*30+N+INT(H/3)-(H=2)+(H=3)-(H>2)*(
H<8)-7*INT(((H-1)*30+N+INT(H/3)-(H=2)+(H
=3)-(H>2)*(H<8))/7))*9+1,9)
```



Ezen a fantasztikus bonyolult címen hirdetjük meg legújabb pályázatunkat. Egyetlen feladatot kell megoldani, s a legjobb megfejtők megkaphatják mindazt, ami az ötletpályázatból megmaradt. Vagyis díjaink:

1 doboz disk

2 db joystick

2 egyéves Ötlet előfizetés

A feladat a következő:

Olvassuk el az ebben a számban megtalálható Sorvezetőt! Beküldendő feladat most a cikk végén található F7-tel jelzett feladat lesz, tehát:

Írjunk BASIC programot (kézzel írott lista is jó), mely valahány számot (a darabszámot az elején kérdezze be a program) sorba rendez az ott közölt összefésülő-program ötlete alapján. Elsősorban arra ügyeljünk, hogy a program sok (pl. 500) szám esetén gyorsan fusson, de azért tartsuk szem előtt a felhasznált memóriát is! A megfelelő leírással kazettán (lemezen) érkező programok közül a legjobbakat (géptípusonként) esetleg közöljük is.

NYERŐ

Kérjük levágni és a levélre felragasztani! Beküldési határidő: 1987. július 31.

BIT-LET Szerkesztőségünk címe: ÖTLET Budapest 1986. Szerkesztő: Angyalosi László. Művészeti szerkesztő: Pribelszky Pál Tördelő: Domokos Imre. Hírrovat: Csányi György. Vallató: Pogány György. Gépnyerő: Király Zoltán. Posta: Halász Péter